The Evolution Towards Grids:

Ten Years of High-Speed, Wide Area, Data Intensive Computing

William E. Johnston

Information and Computing Sciences Division, Lawrence Berkeley National Laboratory

and

NAS Division, NASA Ames Research Center

Abstract

Modern scientific computing involves organizing, moving, visualizing, and analyzing massive amounts of data from around the world, as well as employing large-scale computation. The distributed systems that solve large-scale problems will always involve aggregating and scheduling many resources. Data must be located and staged, cache and network capacity must be available at the same time as computing capacity, etc. Every aspect of such a system is dynamic: locating and scheduling resources, adapting running application systems to availability and congestion in the middleware and infrastructure, responding to human interaction, etc. The technologies, the middleware services, and the architectures that are used to build useful high-speed, wide area distributed systems, are now being integrated in “Grids” [1]. This paper explores some of the background, current state, and future directions of Grids.

1 Introduction

“Grids” are an approach to building dynamically constructed problem solving environments using distributed and federated, high performance computing and data handling infrastructure that incorporates geographically and organizationally dispersed resources.

The overall motivation for most current “Grid” projects is to enable the resource interactions that facilitate large-scale science and engineering such as high energy physics data analysis, climatology, aerospace systems design, etc.

The vision for a computing, data, and instrument Grid is that it will provide significant new capabilities to scientists and engineers by facilitating *routine* construction of information based problem solving environments. Such Grids will knit together widely distributed computing, data, instrument, and human resources into just-in-time systems that can address complex and large-scale computing and data analysis problems. Examples of such problems in the NASA environment include:

- Coupled, multidisciplinary simulations too large for single computing systems (e.g., multi-component turbomachine simulation);
- Management of very large parameter space studies where thousands of low fidelity simulations explore, e.g., the aerodynamics of the next generation space shuttle in its many operating regimes (from Mach 27 entry into the atmosphere to landing);
- Use of widely distributed, federated data archives (e.g., simultaneous access to metrological, topological, aircraft performance, and flight path scheduling databases supporting a National Air Transportation Simulation system);
- Coupling large-scale computing and data systems to scientific and engineering instruments so that real-time data analysis results can be used by the experimentalist in ways that allow direct interaction with the experiment (e.g. operating jet engines in test cells and aerodynamic studies of airframes in wind tunnels);
- Augmented reality and virtual reality remote collaboration (e.g., the Ames / Boeing Remote Help Desk that will provide aircraft field maintenance personnel use of coupled video and non-destructive imaging to supply real-time data to a remote, on-line, airframe structures expert who uses this data to index into detailed design databases, and returns 3D internal aircraft geometry imagery to the field for damage assessment);
- Single computational problems too large for any single system (e.g. extremely high resolution rotocraft aerodynamics calculations).

This paper traces some of the evolution of data intensive computing over the past ten years, which in the opinion of the author, is an elements of Grids equally important with computing. The technology evolution is traced through a series of milestones that are based on advances in the technology, architectures, and software, and that have brought us from the point when we were lucky to get a few hundred kilobits/second of *application-to-application* data transfer on a local area network, to the current time, where we can routinely get 600 megabits/second of data throughput on wide area networks.

This paper also represents a personal, and not a comprehensive, review of the field, though the author has been involved in many of the seminal activities. A number of people will be acknowledged in the course of this article, but there will be those whose important contributions did not directly intersect the author’s work and/or the work of the collaborators, and whom will therefore not be mentioned only for that reason. The body of this paper is organized into three major sections: where are we today, how did we get there, and where are we going in the future.

2 Where Are We Today?

For the current situation, two areas are examined: data intensive computing and NASA's Information Power Grid ([2], [3]). The first of these areas represents important, baseline technology for widely distributed, high performance computing, and the second area is a top-down architecture and implementation project addressing building a prototype production Grid.

2.1 High-Speed, Data Intensive Distributed Systems

As part of a feasibility study for remote access to terabyte sized, high energy physics data sets, experiments were conducted in 1997-98 (see [4] and [5]) between Lawrence Berkeley National Laboratory (LBNL) in Berkeley, Calif., and the Stanford Linear Accelerator (SLAC) in Palo Alto, Calif. The National Transparent Optical Network testbed (NTON - see [6]) testbed provides eight 2.5 gigabit/sec data channels around the San Francisco Bay, of which four are usually used for OC-48 (2.5 gigabit/sec) SONET. For this experiment, the network configuration involved four to six ATM switches and a Sun Enterprise-4000 SMP as a data receiver at SLAC, all with OC-12 (622 Mbit/sec) network interfaces, and four smaller systems at LBNL configured as distributed caches and serving as data sources. The results of this experiment were that a sustained 57 megabytes/sec of data were delivered from datasets in the distributed cache to the remote application memory, ready for analysis algorithms to commence operation. Brian Tierney, who managed these experiments, recently reported in a personal communication, that similar experiments in late 1999 between LBNL and Sandia National Lab, Livermore, CA, that using IP routers with OC-12 packet over SONET interfaces, consistently do even better than this.

This fairly impressive experiment is the result of a ten-year evolution of computing and networking technology, involving advances in platform and network interface technologies, monitoring and management approaches, and parallel distributed software architectures and algorithms.

2.2 NASA's Information Power Grid

Computational Grids, e.g. NASA's Information Power Grid ("IPG"), will provide significant new capabilities to scientists and engineers by facilitating the solution of large-scale, complex, multi-institutional / multi-disciplinary, data and computational based problems using CPU, data storage, instrumentation, and human resources distributed across the NASA community. This entails technology goals of:

- Independent, but consistent, tools and services that support various programming environments for building applications in widely distributed environments

- Tools, services, and infrastructure for managing and aggregating dynamic, widely distributed collections of resources - CPUs, data storage / information systems, communications systems, real-time data sources and instruments, and human collaborators
- Facilities for constructing collaborative, application oriented workbenches / problem solving environments across the NASA enterprise based on the IPG infrastructure and applications. These constitute the primary science and engineering interface to Grids
- A common resource management approach that addresses, e.g., system management, user identification, resource allocations, accounting, security, etc.
- An operational Grid environment incorporating major computing and data resources at multiple NASA sites in order to provide an infrastructure capable of routinely addressing larger scale, more diverse, and more transient problems than is possible today.

What will IPG facilitate?

The goals of an environment with the characteristics noted above will enable NASA scientists to make strides in four classes of activities. First, it will allow for the construction and management of dynamic systems such as wide area testbeds and dynamically configured production environments.

Second, it will allow NASA to prototype distributed systems that can adapt to future changes by using Grid services to flexibly manage changing environments, infrastructure, and resources.

Third, research teams will be able to construct just-in-time, large-scale systems to support scientific and engineering computing and data based activities that are not steady state, i.e. those that may require a different resource mix for every different problem. For example, simulations and their supporting computing platforms including data mining systems and their underlying data archives, instrumentation systems and human collaborators

Finally, IPG will enable the routine use of wide area, data-intensive applications such as those involving remote access to high data-rate real-time data sources and instruments and large datasets as in the on-line medical instrument project [7] mentioned below.

How will IPG be accomplished?

Three main areas must be addressed in order to accomplish these goals:

1) new functionality and capability;

2) an operational environment that encompasses significant resources;

3) new services delivery model.

In the first area, Grids must provide services supporting uniform and location independent interfaces for aggregating, scheduling, and integrating numerous, diverse, and distributed resources.

Such services include resource description and discovery mechanisms; multi-party, secure, and fault tolerant communication; access control; data location management; job submission and data archive access; sharing mechanisms to support collaborative interfaces and toolkits for building problem solving environments.

Some of these services exist and some must be designed, built, and evaluated. These services will knit together and provide access to the many compute and data engines and scientific instruments that will provide significantly increased levels of computing and data analysis capability.

The second area, an operational system, is discussed below.

In the third area, Grids, such as IPG, effectively define a new business model for operational organizations delivering large-scale computing and data resources. Grids require that these services be delivered in ways that allow them to be integrated with other widely distributed resources controlled, e.g., by the user community. This is a big change for, e.g., traditional supercomputer centers. Implementing this service delivery model requires two things: First, tools for production support, management, and maintenance, of integrated collections of widely distributed, multi-stakeholder resources must be identified, built, and provided to the systems and operations staffs. Second, organizational structures must be evolved that account for the fact that operating Grids is different than operating traditional supercomputer centers, and management and operation of this new shared responsibility service delivery environment must be explicitly addressed.

What is the State of IPG?

Point 1), above, is being addressed by a detailed examination of requirements generated by several NASA application communities, both in terms of specific capabilities identified by the applications community, and as the result of analysis of the requirements and desired operating environments by computer scientists. These requirements are documented in [2] and [3].

Addressing point 2), the two year (01/2001) IPG goal is an operational and persistent, “large-scale” prototype-production Information Power Grid providing access to computing, data, and instrument resources at NASA Centers around the country so that applications that cannot be done today are enabled.

The first major milestone toward this goal (01/2000, see [3]) is a baseline Grid system (Figure 1) that

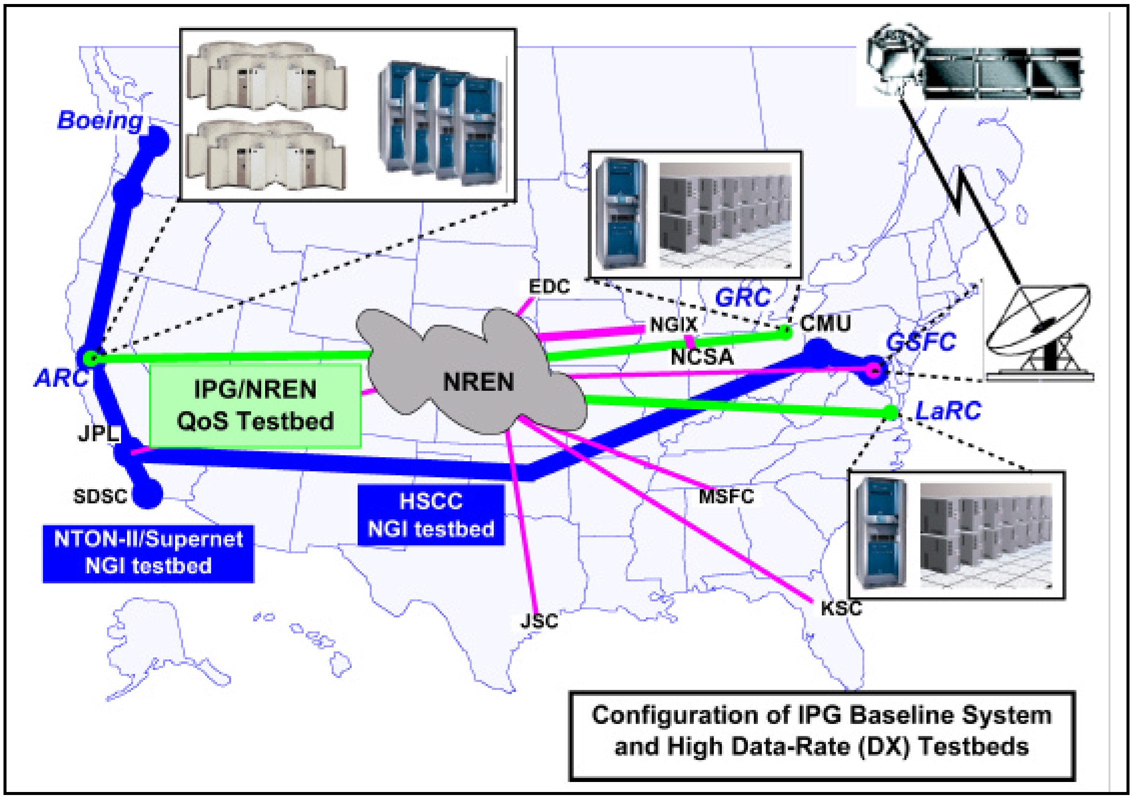

Figure 1 **First Phase of NASA's Information Power Grid**

includes:

- approximately 600 CPU nodes in half a dozen SGI Origin 2000s at three or four NASA sites
- several workstation clusters
- 30-100 Terabytes of uniformly accessible mass storage
- wide area network interconnects of at least 100 mbit/s
- a stable and supported operational environment

Addressing point 3), the NAS Division at NASA Ames is identifying the new services that will be delivered by IPG, and is creating groups that will develop (as necessary), test, deploy, and support these services. In addition to new local organizational structure and local R&D, NAS is coordinating related activities at the NSF supercomputer centers [36], and at several universities, to provide various components of the new operational model.

Current progress is reflected in the IPG Engineering Working Group tasks (see [3]): 30+ tasks have been identified as critical for the baseline system. Task groups are working on each of these, and they fall into the general areas of:

- Identification and testing of computing and storage resources for inclusion in IPG
- Deployment of Globus ([8]) as the initial IPG runtime system
- Global management of CPU queues, and job tracking, and monitoring throughout IPG
- Definition and implementation of a reliable, distributed Grid Information Service that characterizes all of the IPG resources
- Public-key security infrastructure integration and deployment to support single sign-on using X.509 cryptographic identity certificates (see [9])
- Network infrastructure and QoS
- Mass storage system metadata catalogue and uniform access system (based on MCAT/SRB - [10])
- Operational and system administration procedures for the distributed IPG
- User and operations documentation
- Account and resource allocation management across systems with multiple stakeholders
- Grid MPI [11], CORBA [13], and Legion [12] programming middleware systems integration
- High throughput job management tools
- Distributed debugging and performance monitoring tools

3 How Did We Get Here?

In the next several sections I will relate some of the evolutionary steps, then return to the concept of Grids in the final section. (See [14] for more complete descriptions of several of these examples.)

3.1 The Gore Demonstration: Selling the Potential

In the spring of 1989, then Senator Al Gore was holding hearings on his High Performance Computing and Communication legislation. At one of the early hearings, Craig Fields, then head of DARPA, was invited to provide testimony on the impact of high-speed networks. Through various circumstances, LBL was asked to provide a demonstration that would relate remote visualization and networking. A “live” network connection was ruled out (we were told that this exercise was the first computer demonstration in a Senate hearing room and they did not want to try for a network connection on top of everything else) so a realistic “simulation” was required. A collection of scientific visualization movies were put together, and, at the suggestion of Mark Pullen (DARPA), Steve Casner (then of ISI) and Van Jacobson (LBL) did various measurements on the new NSFNet T3 (45 megabits/sec) Internet backbone. They measured packet delays on cross country connections, and those delays were then used to clock out the movie frames for display on a graphics workstation to simulate transmission of the movie frames across networks of various speeds (19 kilobits/sec to 40 megabits/sec). The resulting video display of the movie gave the Senators an appreciation for implications of data network bandwidth. The Federal HPCC program that grew out of this hearing has provided funding for many of the projects described here.

3.2 Supercomputing 1991: Demonstrating the Potential

A demonstration at SC91 (in Albuquerque, NM) was arguably the first use of “high-speed” wide area networks to support a high-speed TCP/IP based distributed application.

The goal was to demonstrate real-time remote visualization of a large, complex scientific dataset. The approach was to use a Thinking Machines’ CM-2 and Cray Y-MP at the NSF’s Pittsburgh Supercomputer Center (PSC) to compute the visualization of a large medical dataset (a high-resolution MRI scan of a human brain). This type of data is essentially a 3D scalar field, and contours of this data represent surfaces of various types of brain tissue and structures. It is these surfaces that are identified and displayed. This visualization involved a CM-2 and Cray Y-MP at the Pittsburgh Super Computer Center producing graphics at 10-12 frames/sec. These frames were sent over the NSFNet 45 megabit/sec connection to the SC91 exhibits. The details may be found in [14] and [15].

Typical of distributed applications, many components had to interoperate to produce a functioning system, an especially difficult task in a wide-area network. David Robertson and Brian Tierney (LBL), and Wendy Huntoon, Jamshid Mahdavi, and Matt Mathis (PSC) spent a lot of time getting the CM-2, the Cray, and the network to interoperate. Dave Borman (Cray) and Van Jacobson (LBL) were doing kernel hacking on the Cray and the Sun up to the hour that the SC91 exhibits opened in order to accomplish the first heterogeneous operation of the TCP large window option that made the 15 Mbit/sec TCP possible between Albuquerque and Pittsburgh. (Less than 2 Mbit/sec were possible using the standard 64-kilobyte TCP windows).

The enduring legacy of this work was the experience gained in building widely distributed systems and the TCP modifications that allowed high data rates in the wide area.

3.3 BAGNet: Involvement of a Large Community and, Finally, a “Real” Application

BAGNet was an IP over OC-3 (155 Mbit/sec) ATM, metropolitan area testbed that operated in the San Francisco Bay Area (California) for two years starting in early 1994. The participants included government, academic, and industry computer science and telecommunications R&D groups from fifteen

Bay Area organizations. The goal was to develop and deploy the infrastructure needed to support a diverse set of distributed applications in a large-scale, IP-over-ATM network environment. The participating organizations were Apple Computer, DEC – Palo Alto Systems Research Center, Hewlett-Packard Laboratories, International Computer Science Institute, Lawrence Berkeley National Laboratory, Lawrence Livermore National Laboratory (LLNL), NASA Ames Research Center, Pacific Bell – Broadband Development Group, Sandia National Laboratories, Silicon Graphics, Inc., SRI International, Stanford University, Sun Microsystems, Inc., University of California, Berkeley, and Xerox Palo Alto Research Center (PARC).

The testbed consisted of a full-mesh, unicast ATM/PVC network supporting four end nodes at each of the fifteen sites, and a full-mesh ATM point-to-multipoint (multicast) link structure for each of the 15 sites. The unicast mesh provided an ATM "best-effort" quality of service over a 155-Mbit/sec SONET infrastructure between the (approximately) 60 connected systems. A single logical IP subnet overlaid this ATM network supporting a variety of distributed applications – see, for example, [16]. The ATM point-to-multipoint mesh was used to support IP multicast, and this capability supported high-quality multimedia teleseminars using the MBone tools: vic, vat, and wb. [17]

The PVC mesh consisted of about 1800 virtual circuits - a herculean management task without the tools available today, that was accomplished by Berry Kercheval of Xerox PARC. The interior (central office) switches were primitive, and the whole network worked poorly until Lance Berc of DEC's Systems Research Center, Helen Chin of Sandia Livermore, and Dave Wiltzius of Lawrence Livermore National Lab identified a set of key central office ATM switch issues that Pacific Bell could address. (See [16].)

In addition to "community" projects in BAGNet, there were several specific projects involving subsets of the connected sites. In particular, LBNL, the Kaiser Permanente health care organization, and Philips Palo Alto Research Center collaborated to produce a prototype production, on-line, distributed, high data rate medical imaging system. (Philips and Kaiser were added to BAGNet for this project through the Pacific Bell CalREN program.)

The Kaiser project ([18]) focused on using high data rate, on-line instrument systems as remote data sources. What was learned in this project was that when data is generated in large volumes and with high throughput, and especially in a distributed environment where the people generating the data are geographically separated from the people cataloguing or using the data, there are several important considerations:

- automatic generation of at least minimal metadata;

- automatic cataloguing of the data and the metadata as the data is received (or as close to real time as possible);
- transparent management of tertiary storage systems where the original data is archived;
- facilitation of cooperative research by providing specified users at local and remote sites immediate as well as long-term access to the data;
- mechanisms to incorporate the data into other databases or documents.

The WALDO (Wide-Area Large-Data-Object) system was developed to provide these capabilities, especially when the data is gathered in real time from a high data rate instrument. WALDO is a digital data archive that is optimized to handle real-time data. It federates textual and URL linked metadata to represent the characteristics of large data sets. Automatic cataloguing of incoming real-time data is accomplished by extracting associated metadata and converting it into text records; by generating auxiliary metadata and derived data; and by combining these into Web-based objects that include persistent references to the original data components (called large data objects, or LDOs). Transparent, tertiary storage management for the data components (i.e., the original datasets) is accomplished by using the remote program execution capability of Web servers to manage the data on mass storage systems. For subsequent use, the data components may be staged to a local disk and then returned as usual via the Web browser, or, as is the case for several of our applications, moved to a high-speed cache for access by specialized applications (e.g., the high-speed video player illustrated in the right-hand part of the right-hand panel in Figure 2). The location of the data components on tertiary storage, how to access them, and other descriptive material are all part of the LDO definition. The creation of object definitions, the inclusion of “standardized” derived-data-objects as part of the metadata, and the use of typed links in the object definition, are intended to provide a general framework for dealing with many different types of data, including, for example, abstract instrument data and multi-component multimedia programs. See [18].

WALDO was used in the Kaiser project to build a medical application that automatically manages the collection, storage, cataloguing, and playback of video-angiography data[*] collected at a hospital remote from the referring physician.

Using a shared, metropolitan area ATM network and a high-speed distributed data handling system, video sequences are collected from the video-angiography imaging system, then processed, catalogued, stored, and made available to remote users. This permits the data to be made available in near-real time to remote

[*] Cardio-angiography imaging involves a two plane, X-ray video imaging system that produces from several to tens of minutes of digital video sequences for each patient study for each patient session. The digital video is organized as tens of data-objects, each of which are of the order of 100 megabytes.

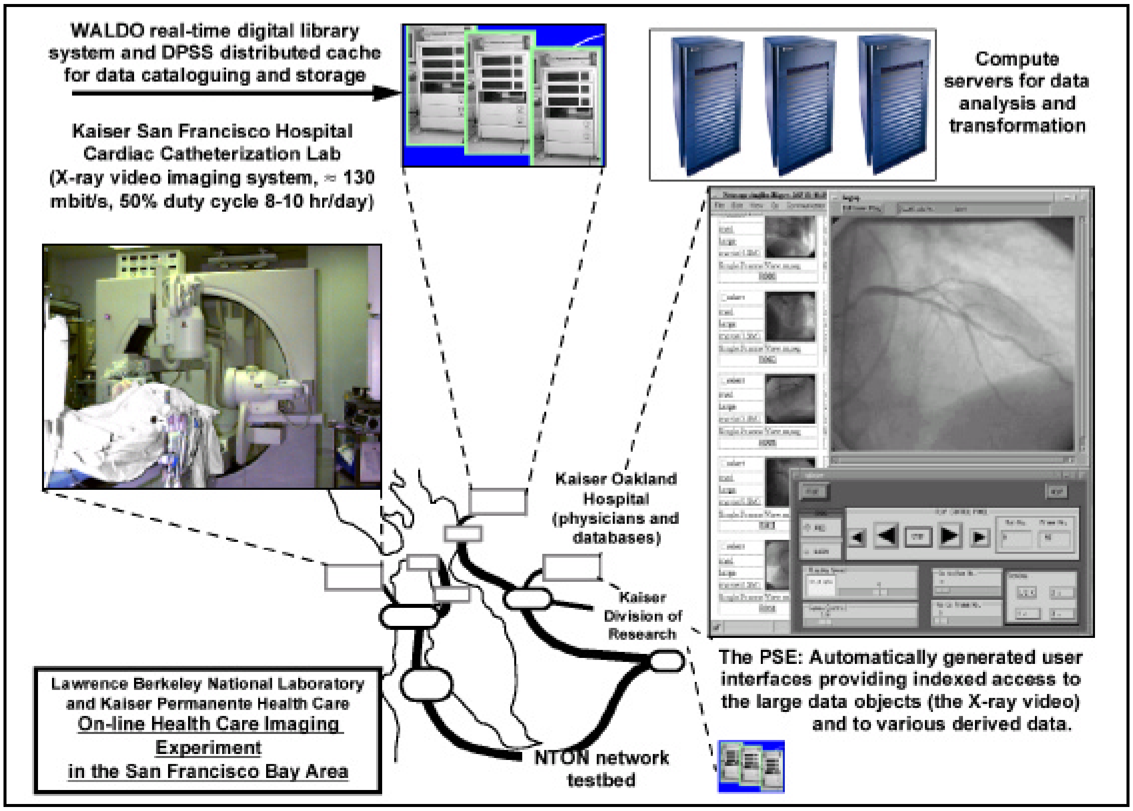
WALDO real-time digital library system and DPSS distributed cache for data cataloguing and storage
Compute servers for data analysis and transformation
Kaiser San Francisco Hospital Cardiac Catheterization Lab (X-ray video imaging system, ≈ 130 mbit/s, 50% duty cycle 8-10 hr/day)
Kaiser Oakland Hospital (physicians and databases)
Kaiser Division of Research
The PSE: Automatically generated user interfaces providing indexed access to the large data objects (the X-ray video) and to various derived data.
Lawrence Berkeley National Laboratory and Kaiser Permanente Health Care
On-line Health Care Imaging Experiment in the San Francisco Bay Area
NTON network testbed

Figure 2 An On-Line Instrument Managed by a Distributed System

clinics (see Figure 2). The LDO becomes available as soon as the catalogue entry is generated — derived data (e.g. MPEG versions of the instrument digital video) is added as the processing required to produce it completes. Whether the storage systems are local or distributed around the network is entirely a function of optimizing logistics.

In the Kaiser project, cardio-angiography data was collected directly from a Philips scanner by a computer system in the San Francisco Kaiser hospital Cardiac Catheterization Laboratory. This system is, in turn, attached to an ATM network provided by the NTON and BAGNet testbeds. When the data collection for a patient is complete (about once every 20–40 minutes), 500–1000 megabytes of digital video data is sent across the ATM network to LBNL (in Berkeley) and stored first on the DPSS ([19]) distributed cache (described below), and then the WALDO object definitions are generated and made available to physicians in other Kaiser hospitals via BAGNet. Auxiliary processing and archiving to one or more mass storage systems proceeds independently. This process goes on 8–10 hours a day.

WALDO provides the Web-based user interface to the data and to appropriate viewing applications. Hospital department-level Web-based patient databases can then refer directly to the data in WALDO without duplicating that data, or being concerned about tertiary storage management (which is handled by WALDO).

The legacy of this project for data intensive environments is a general model for data intensive computing. See [14].

3.4 MAGIC: the First “Real” Data Intensive Environment

The MAGIC Gigabit testbed* was a DARPA-funded collaboration working on distributed applications in large-scale, high-speed, ATM networks. It was a heterogeneous collection of ATM switches and computing platforms, several different implementations of IP over ATM, a collection of “middleware” (distributed services), etc., all of which must cooperate in order to make a complex application operate at high speed.

Another key aspect of a data intensive computing environment has turned out to be a high-speed, distributed cache for managing and providing high speed access to large data sets in widely distributed environments. LBNL designed and implemented the Distributed-Parallel Storage System (DPSS - [19]) as part of the MAGIC project, and as part of the U.S. Department of Energy’s high-speed distributed computing program. This technology has been quite successful in providing an economical, high-performance, widely distributed, and highly scalable architecture for caching large amounts of data that can potentially be used by many different users. In the MAGIC testbed a multi-server DPSS was typically distributed across several sites separated by more than 2600 km of high-speed, IP-over-ATM network, and is used to store very high resolution images of several geographic areas (see Figure 3). The first application use of the DPSS was *TerraVision*, a terrain visualization application that uses the DPSS to let a user explore / navigate a “real” landscape represented in 3D by using ortho-corrected, one meter per pixel images and digital elevation models (see [22]). TerraVision requests from the DPSS, in real time, the sub-images (“tiles”) needed to provide a view of a landscape for an autonomously “moving” user. Typical use requires aggregated data rates as high as 100 to 200 Mbits/sec. The DPSS was easily able to supply these data rates from several disk servers distributed across the network.

A central issue for using high-speed networks and widely distributed systems as the foundation of a large data-object management strategy is the performance of the system components, the transport and OS software, and the underlying network. Problems in any of these regimes will hinder a data intensive computing strategy, but such problems can usually be corrected if they can be isolated and characterized.

*MAGIC was established in June 1992 by the U. S. Government’s Defence Advanced Research Projects Agency (DARPA), and operated until mid-1999. The testbed was a collaboration between LBNL, Minnesota Supercomputer Center, SRI, Univ. of Kansas, Lawrence, KS, USGS - EROS Data Center, CNRI, Sprint, and Splitrock Telecom. See [20] and [21].

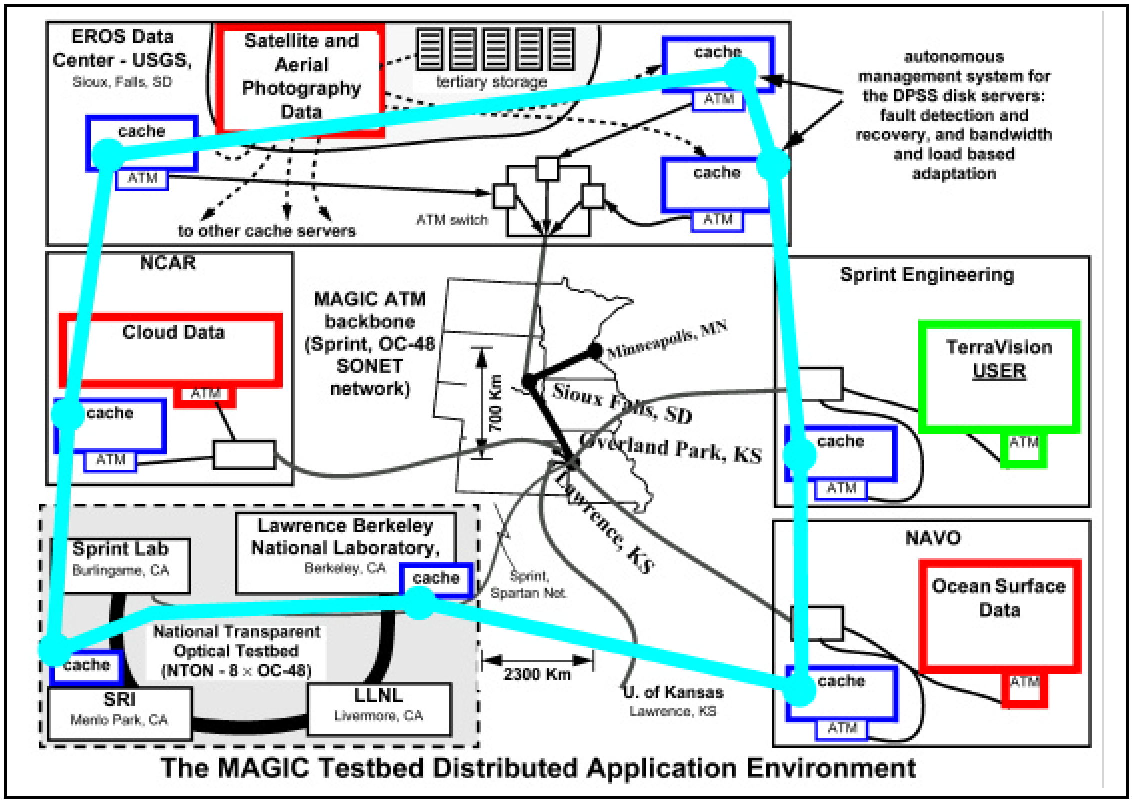

Figure 3 Aspects of the MAGIC Project Focused on Autonomous Management of Widely Distributed Components

The DPSS serves several roles in high-performance, data-intensive computing environments. This application-oriented cache provides a standard high data rate interface for high-speed access by data sources, processing resources, mass storage systems (MSS), and user interface elements. It provides the functionality of a single very large, random access, block-oriented I/O device (i.e., a “virtual disk”) with very high capacity (we anticipate a terabyte sized system for high-energy physics data) and serves to isolate the application from tertiary storage systems and instrument data sources. Many large data sets may be logically present in the cache by virtue of the block index maps being loaded, even if the data is not yet available. In this way processing can begin as soon as the first data blocks are generated by an instrument or migrated from tertiary storage.

The DPSS is a collection of wide area distributed disk servers that operate in parallel to provide high-speed logical block level access to large, named data sets. These data sets are broken up into 64 KB blocks that are declustered (dispersed in such a way that as many system elements as possible can operate simultaneously to satisfy a given request) across both disks and servers. This strategy allows both a large collection of disks to seek in parallel, and all servers to operate in parallel to send the requested data to the application, enabling the DPSS to perform as a high-speed data block server.

Building reliable, large-scale, widely distributed applications and systems is, as much as anything else, an exercise in adaptive monitoring and semi-autonomous management of the system components and job elements.

Independent monitoring is necessary because remote application and system components may cease responding for either normal or abnormal reasons and are typically not easily accessible for diagnosis. Job elements can crash, system components operating in widely distributed environments can crash, networks can partition, local resource usage agreements can be abrogated, etc. The monitors themselves can fail and this must be detectable and correctable without direct human intervention.

Autonomy is needed in order to deal with systems that have lost contact with central administrators or have to deal with problems on time or size scales not addressable by human administrators.

Adaptability is needed in order to accommodate experiments, new configuration and monitoring requirements, etc., without having to reinstall the management system on all of the widely distributed hosts where is might be running.

Both to illustrate the issues and to describe one approach that has been demonstrated to be feasible and effective, we will describe a research prototype that indicates how some of the issues noted above may be addressed.

Autonomous System Management

The JAMM system ("Java Agents for Monitoring and Management", see [23]), developed at LBNL by Chris Brooks and Brian Tierney, uses Java-based agents and brokers to control and monitor a complex, highly distributed parallel application (the WALDO real-time digital library system [18]). The importance of this work is not just the approach – which has proved very successful – but also the fact that the system was implemented, installed, debugged, and refined in a realistic wide area distributed application.

Internally the DPSS is itself a distributed system consisting of a name server, where the logical block names are translated to physical addresses (server: disk: disk offset), and several disk servers. In the disk servers, the data is read from disk into local cache, and then sent to the applications. DPSS configurations involving a dozen independent components have been maintained where they span separate geographic and administrative domains. These experiences have demonstrated the difficulties in configuring and maintaining widely distributed systems by hand.

In the JAMM approach, monitoring and management agents use high-level reasoning to solve problems and perform automated tasks related to managing remote system state. The tasks are defined as Java classes, and can thus be loaded into monitor agents remotely. The agents use reliable IP-multicast to communicate with other associated agents, which provides an independent monitoring of the agents themselves. These agents are also useful for performance monitoring of distributed systems.

Brokers can be used to keep track of multiple, independent DPSS systems. Clients may wish to replicate their data across multiple DPSS's and choose between them depending on the location of the client at a given time. Brokers know which data sets are loaded on each DPSS and which DPSS has the best network connectivity to a given client, and can advise the client which is the best DPSS to use.

When new components, such as a new disk server, are added to a DPSS configuration, the agents do not have to be reconfigured or restarted. When an agent is started on the new host it will inform all other agents about the new server. Agents are able to continually propagate information about the state of the system to each other, such as the addition and deletion of hosts, disks and interfaces.

New agent methods (functions) can be added at any time. Agents are capable of informing each other about new tasks to be performed or about changes in existing tasks. For example, the brokers have an algorithm for determining which DPSS configuration to use based on a set of parameters that include network bandwidth, latency and current server load. If desired, this algorithm may be modified “on the fly” without having to reinstall a new set of binaries on every host in the system, because the agents can accept downloaded code. A DPSS administrator, for example, can also design a new task that automatically moved all data sets on a server the were more than one week old to tertiary storage. Once this was “taught” to one agent, that agent will then propagate the task to all the other agents in the agency, saving the administrator from needing to manually reconfigure each agent.

An agent can also be used to assist in the management of data sets. For example, an agent can be instructed to replicate data sets across a second set of disks, automatically compress image sets, migrate sets to mass storage after a given time or keep track of the content of an evolving data set.

To be specific, “WHERE” is an implementation of this approach. WHERE agents are used to monitor and restart DPSS servers and to collect and display near-real-time statistics about a DPSS. The organization of agents to accomplish this sort of thing is shown in Figure 4. Note that the DPSS servers were scattered all over the country in the MAGIC testbed project.

In a research environment, DPSS servers typically have multiple networks interfaces (i.e.: ethernet, FDDI, and ATM), one or more of which may be down at any given moment. The agents are used to test all possible network interfaces every few minutes, and keep track of the fastest available interface on each server at any given moment. When a DPSS client requests a data set from a DPSS, it queries the broker associated

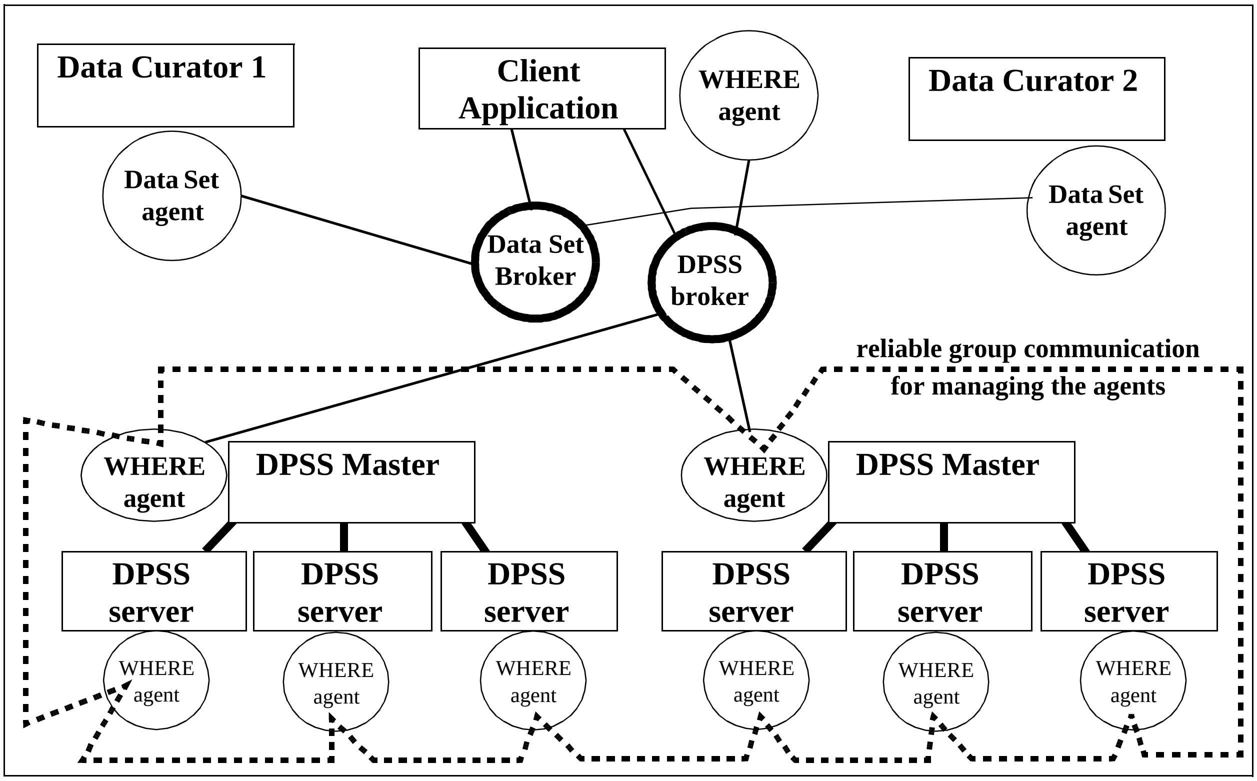

Figure 4 **DPSS Use of Agents and Brokers**

"Where" agents monitor the state of the sub-system components to supply state information and to restart failed components. The data set brokers and agents manage data migration to and from the DPSS cache. The DPSS servers were scattered all over the country in the MAGIC testbed project.

with that DPSS's agency to find the fastest interface (Ethernet, ATM, etc.) between the client and server machines. Without the agents, the clients would need to try all possible interface to determine the fastest, a process that can take several seconds, or even longer in a wide-area ATM environment where call setup time may take up to one minute per connection.

WHERE agents are also able to continuously monitor DPSS servers. When an agent detects that a server has failed, it attempts to restart it. If it is unable to restart the server, it can take an alternate action, such as sending email to the administrator of this DPSS informing them of the problem. This autonomy makes the DPSS much more robust and eases the duties of a DPSS administrator.

Applications operating in a distributed environment within a wide-area network often need knowledge about the state of the network and the hosts within it in order to operate efficiently. However, much of this knowledge, such as the latency between two remote hosts or the number of users on a particular server, is not easily determined from a single node within the network. Furthermore, when things go wrong with the network or the servers, one frequently cannot access the state of the system after the fact -- some sort of continuous state monitoring is needed to maintain a global history of the system. Additionally, it is desirable for this monitoring system to be generalized and adaptable: new types of users or applications may have different monitoring requirements. Analysis of the events leading to congestion or system failure may also require accessing different types of data at different granularities than those examined during normal operation, thus requiring adaptability of the monitors. The system must also have a great deal of autonomy. It must be able to make decisions and execute tasks without prompting a human user.

WHERE agents and brokers are also able to provide a near-real-time picture of the DPSS, its clients, and the relative usage of various servers and network connections. The broker collects these statistics from agents throughout the system, collates and massages them and forwards them to an applet that provides users with a graphical representation of the system. Previously these statistics were available, but difficult to access in real time. A user could determine after the fact how the system had changed by using the logs and graphs generated from them, but had little means of watching the effects of an experiment on the system while the experiment was taking place. WHERE provides DPSS users with that capability.r

To reiterate, these agents are autonomous, adaptable entities that are capable of filtering information about the state and history of a system, such as the throughput between servers, the number of users in a system, or the location of different copies of a given data set. They are also able to maintain a continually updated view of the global state of the system, which allows users to optimize the configuration used depending on their particular requirements. In addition, they are able to independently perform sets of administrative tasks, such as the restarting of server processes.

In summary, agents have proved to be virtually essential in monitoring and managing highly distributed systems. They allow users and administrators to maintain a global view of the system, even if the components are geographically and administratively separate. WHERE’s high-level knowledge-based design, can determine a course of action, rather than hard coding actions into the agents, allows the agents to grow and adapt to the management oriented monitoring and active management of distributed systems.

The legacies of DPSS in the MAGIC testbed for data intensive computing are the establishment of the importance of a high-speed distributed cache, the agent based monitoring and management architecture, a highly distributed security architecture, and the development of “global” data management strategies.

4 Where We Are Today, Revisited: An Overall Model for Grids

Analysis of some specific requirements ([3]), and of the work processes of the user communities, as well as some anticipation of where the technology and problem solving needs are going in the future leads to a characterization of the desired Grid functionality. This functionality may be described as a hierarchically structured set of services and capabilities.

Problem Solving Environments, Supporting Toolkits, and High-Level Services

A number of services directly support building and using the Grid problem solving environment, e.g., by engineers or scientists. These include the toolkits for construction of application frameworks / problem solving environments (PSE) that integrate Grid services and applications into the "desktop" environment. For example, the graphical components ("widgets" / applets) for application user interfaces and control; the computer mediated, distributed human collaboration that support interface sharing and management; the tools that access the resource discovery and brokering services; tools for generalized workflow management services such as resource scheduling, and managing high throughput jobs, etc.

An important interface for developers of Grid based applications is a "global shell," which, in general, will support creating and managing widely distributed, rule-based workflows driven from a published / subscribed global event service. Data cataloguing and data archive access, security and access control are also essential components. The PSE must also provide functionality for remote operation of laboratory / experiment / analytical instrument systems, remote visualization, and data-centric interfaces and tools that support multi-source data exploration.

Programming Services

Tools and techniques are needed for building applications that run in Grid environments, cover a wide spectrum of programming paradigms, and must operate in a multi-platform, heterogeneous computing environments. NASA's IPG, e.g., will require Globus support for Grid MPI [11] as well as Java bindings to Globus services. CORBA [13], Condor [24], Java/RMI [25], Legion [12], and perhaps DCOM [26]. Compilation environment management, distributed debugging and performance analyses are difficult and important areas that must also be addressed.

Tools are needed for converting and "wrapping" legacy codes for operation in Grids, and for incorporating legacy Fortran codes into CORBA environments that are used to support composing application components. Grid-enabled numerical solution libraries that can be optimized for distributed architectures are important, as are services such as NetSolve (http://www.cs.utk.edu/netsolve/).

Grid Common Services: Execution Management

Several services are critical to managing the execution of application codes in the Grid. The first is resource discovery and brokering. By discovery we mean the ability to ask questions like: how to find the set of objects (e.g. databases, CPUs, functional servers) with a given set of properties; how to select among many possible resources based on constraints such as allocation and scheduling; how to install a new object/service into the Grid; and how make new objects known as a Grid service. The second is execution queue management, which relates to global views of CPU queues and their user-level management tools. Workflow management and global shells is the third category. The fourth category is distributed application management. The last category includes tools for generalized fault management including multi-level autonomous management mechanisms for system components and applications and process monitoring and supplying information to knowledge based recovery systems.

Grid Common Services: Runtime

Globus has been chosen as the initial IPG runtime system and supplies basic services to characterize and locate resources, initiate and monitor jobs, and provide secure authentication of users. However, there are other runtime services that are needed, include checkpoint/restart mechanisms, access control, a global file system, and Grid communication libraries such as a network-aware MPI that supports security, reliable multicast and remote I/O.

High-speed, wide area, distributed data management services include global naming and uniform access, uniform naming and location transparent access to resources such as data objects, computations, instruments and networks that work through Grid-wide object brokers. This, in turn requires uniform I/O mechanisms (e.g. read, write, seek) for all access protocols (e.g. http, ftp, nfs, Globus gass...) and richer access and I/O mechanisms (e.g. “application level paging”) that are present in existing systems.

Data cataloguing and publishing constitute another needed class of services. These include the ability to automatically generate the meta-data about data formats, and management of use conditions and access control. The ability to generate model based abstractions for data access using extended XML and XMI [27] data models is also likely to be important in the complex and data rich environment of, e.g., aero-space design systems.

Of course, high-speed, wide area, access to tertiary storage systems will always be critical, in the science and engineering applications that we are addressing. In IPG we are using SDSC’s Meta Data Catalogue / Storage Resource Broker (“MCAT/SRB”) [10] to provide widely distributed access to tertiary storage systems, independent of the nature of the underlying mass storage system implementation.

High-performance applications require high-speed access to data files, and the system must be able to stage, cache, and automatically manage the location of local, remote and cached copies of files. We are also going to need the ability to dynamically manage large, distributed “user-level” caches and “windows” on off-line data. Support for object-oriented data management systems will also be needed.

Services supporting collaboration and remote instrument control are needed. In addition, application monitoring and application characterization, prediction, and analysis, will be important for both users and the managers of the Grid.

Finally, monitoring services will include precision time event tagging for dispersed, multi-component performance analysis as well as generalized auditing data file history and control flow tracking in distributed, multi-process simulations.

Grid Common Services: Environment Management

The key service that is used to manage the Grid environment is the “Grid Information Service.” This service – currently provided by Globus GIS (formerly MDS, see [28]) – maintains detailed characteristics and state information about all resources, and will also need to maintain dynamic performance information, information about current process state, user identities, allocations and accounting information.

Autonomous system management and fault management services provide the other aspect of the environmental services.

Resource Management for Co-Scheduling and Reservation

One of the most challenging and well known Grid problems is that of scheduling scarce resources such as a large instruments. In many, if not most, cases the problem is really one of co-scheduling multiple resources. Any solution to this problem must have the agility to support transient experiments based on systems built on-demand for limited periods of time. CPU advance reservation scheduling and network bandwidth advance reservation scheduling based on differentiated IP services are critical components to the co-scheduling services. In addition, tape marshaling in tertiary storage systems to support temporal reservations of tertiary storage system off line data and/or capacity is likely to be essential.

Operations and System Administration

Implementing a persistent, managed Grid requires tools for deploying and managing the system software. In addition, tools for diagnostic analysis and distributed performance monitoring are required, as are accounting and auditing tools. An often overlooked service involves the operational documentation and procedures that are essential to managing the Grid as a robust production service.

Access Control and Security

The first requirement for establishing a workable authentication and security model for the Grid is to provide a single-sign-on authentication for all Grid resources based on cryptographic credentials maintained in the users desktop / PSE environment(s) or on one’s person. In addition, end-to-end encrypted communication channels are needed in for many applications in order to ensure data integrity and confidentiality.

The second requirement is an authorization and access control model that provides for management of stakeholder rights (use-conditions) and trusted third parties to attest to corresponding user attributes. A policy-based access control mechanism that is based on use-conditions and user attributes is also a requirement.

Security and infrastructure protection are, of course, essential requirements for the resource owners. This area includes technologies such as IPSec and secure DNS to authenticate IP packet origin, secure router and switch management, etc. (See, e.g., [29].)

Services for Scalability

There are a number of services and design considerations that are necessary to ensure that the Grid will scale numerically, geographically, organizationally, and functionally. Some of the issues involved in scalability are the ability to broker and manage resources and handle faults autonomously; very reliable access to “global” system state information, and; general policy based access control and use-condition management that operates relatively automatically and has distributed management.

Services for Operability

To operate the Grid as a reliable, production environment is a challenging problem. Some of the identified issues include management tools for the Grid Information Service that provides global information about the configuration and state of the Grid; diagnostic tools so operations/systems staff can investigate remote problems, and; tools and common interfaces for system and user administration, accounting, auditing and job tracking. Verification suites, benchmarks, regression analysis tools for performance, reliability, and system sensitivity testing are essential parts of standard maintenance.

Grid Architecture: How do all these services fit together?

We envision the Grid as a layered set of services (see Figure 5) that manage the resources, and middleware that supports different styles of usage (e.g. different programming paradigms).

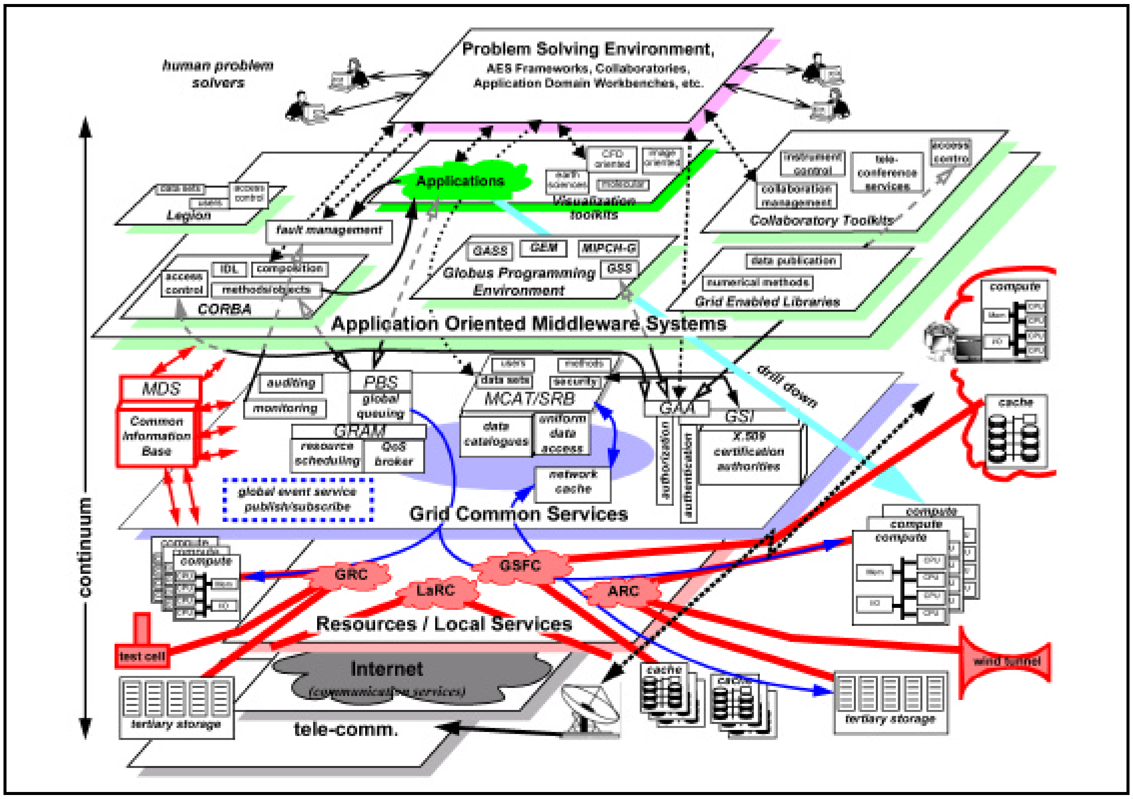

Figure 5 A Representation of a Grid Architecture

However, the implementation is that of a continuum of hierarchically related, independent and interdependent services, each of which performs a specific function, and may rely on other Grid services to accomplish its function.

Further, the “layered” model should not obscure the fact that these “layers” are not just APIs, but usually a collection of functions and management systems that work in concert to provide the “service” at a given “layer.”

The arrows in the figure between several of the layers and services are intended to indicate how a real application involving a team working on a computational fluid dynamics (“CFD”) based design problem might interact with Grid services, top to bottom.

5 Grids: What Comes Next?

The vision is that Grids (and IPG) will routinely – and easily, from the user’s point of view – provide for:

- building just-in-time, large-scale systems/applications to support scientific and engineering computing and data oriented activities that are not steady state, i.e. those that may require, or have to make use of, a different resource mix for every different problem – e.g.:

    - coupled, multidisciplinary simulations
    - large-scale simulations
    - coordinated use of many dispersed data archives
    - data analysis based control of on-line instruments
    - coupling of remote, on-line instruments to large-scale computation simulation
- routine use of wide area, data-intensive applications
    - remote access to high data-rate real-time data sources / instruments and very large datasets
- building and managing just-in-time, dynamic, and collaborative systems
    - coordinated work by dispersed groups based, e.g., on central design databases (e.g. airframe or turbomachinery geometry and performance)
    - data and simulation based crisis response

There are several motivations for this vision. In addition to the large scale aero-space systems design problems that drive NASA's interest in Grids, the U.S. Dept. of Energy operates some of the world's largest, on-line, shared scientific instrument systems. These include high voltage electron microscopes, synchrotron light sources, high energy particle accelerators, gigahertz NMR systems, special telescopes, quartz piston diesel engines, and so on. These instrument systems - many of which are national user facilities - have a wide variety of computing and storage requirements associated with them, but they share certain characteristics that motivate much of our work in data intensive computing. These instruments are used by people in research labs and universities all over the world. It has already been shown in various collaboratory* projects that remote users will gain significant efficiencies through transparent remote access involving widely distributed data intensive computing in almost every case. See, for example, [30], [31], [32], and [33].

However, the major scientific gains are likely to come when we can couple these instruments to large-scale computing and storage systems which is another motivation for the rationalized and persistent computing, storage, and collaboration infrastructure of Grids. In most cases, observation and experimentation advances our knowledge when we are able to match experiment and theory. Today this is dramatically slowed by the long time required to collect data and do off-line processing in order to compare experiments with

*"The fusion of computers and electronic communications has the potential to dramatically enhance the output and productivity of U. S. researchers. A major step toward realizing that potential can come from combining the interests of the scientific community at large with those of the computer science and engineering community to create integrated, tool-oriented computing and communication systems to support scientific collaboration. Such systems can be called 'collaboratories.'" From "National Collaboratories - Applying Information Technology for Scientific Research," Committee on a National Collaboratory, National Research Council. National Academy Press, Washington, D. C., 1993.

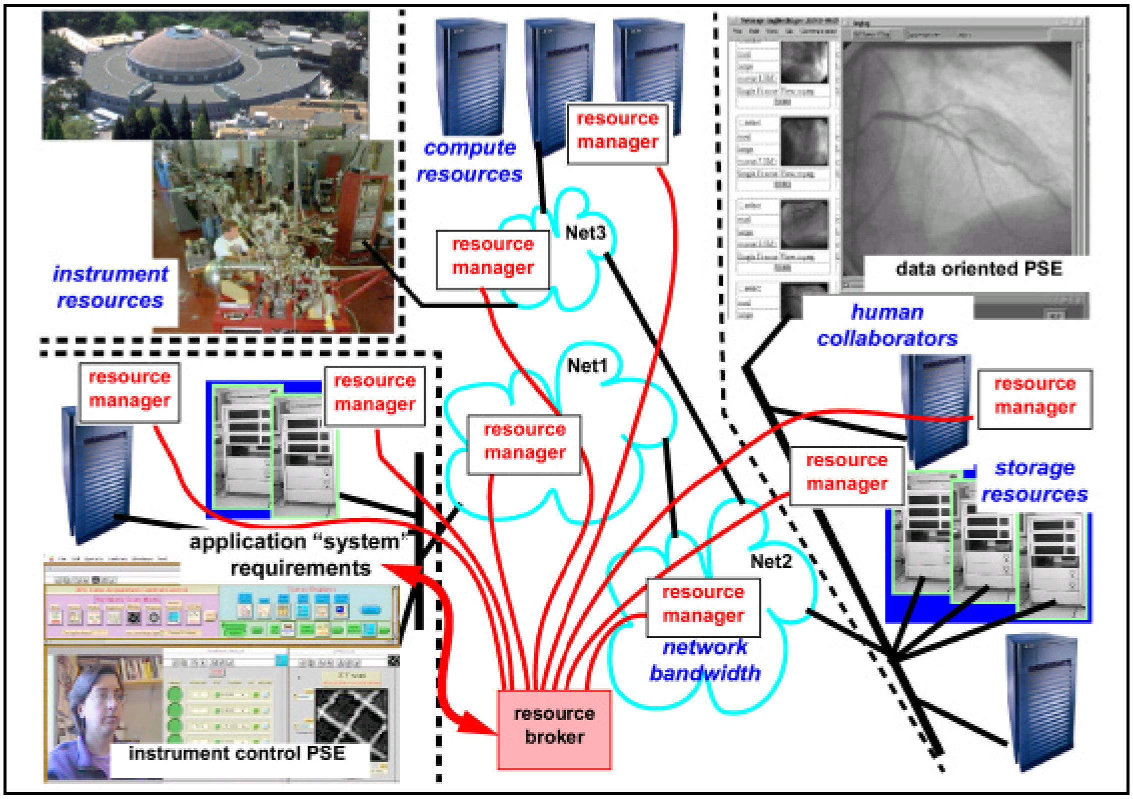

Figure 6 Collaboratories Involve Distributed Instrumentation Systems that Require Coordinating Many Different Resources

computational models that represent the current theory of the underlying physical processes. Both the experiment and the model are adjusted and the process repeated. The process of analyzing experiments and comparing them with theory is almost always hampered by the lack of sufficient storage and computation.

Developing the flexible, transparent, and dynamic data intensive computing environments of Grids will provide a big step forward by simultaneously coupling experiments and instruments to large scale computing, and by providing greater capacity for simulation. Ultimately, the flexibility and transparency of Grids will allow scientific experiments to be directly coupled to the computation simulations of the subject phenomenon through the use of high-speed networks, distributed storage systems and computation that can be scheduled so that all of the required resources are available to match small windows of instrument operation time, etc. When this happens, the weeks of off-line computational analysis of experiment data followed by the (largely manual) feedback of experimental results into the models, and vice versa, should be greatly shortened, permitting more and different kinds of experiments, more accurate and detailed insights, etc.

This vision of cooperative operation of scientific experiments and computational simulation of the theoretical models is one of the ultimate goals for our work in Grids.

In this regard, the challenge addressed by Grids is how to accelerate routine use of these applications that:

- require substantial computing resources
- generate and/or consume high rate and high volume data flows
- involve human interaction
- require aggregating many dispersed resources to establish an operating environment:
    - multiple data archives
    - distributed computing capacity
    - distributed cache capacity
    - “guaranteed” network capacity
- operate in widely dispersed environments.

5.1 The Challenge

There are many challenges in making Grids a reality, in the sense that they can provide new capabilities in production quality environments.

While the basic Grid services have been demonstrated, e.g. in the GUSTO testbed ([34]), a general purpose computing, data management, and real-time instrumentation Grid involves many more services. One challenge is to identify the minimal set of such services. In many cases, these services exist in some form in R&D environments, as described in this paper, however, then the challenge is to convert these into robust implementations that can be integrated with the other services of the Grid. This is hard, and is one of the big challenges for NASA’s IPG.

Fortunately, Grids are being developed by a substantial and increasing community of people who work together in a loosely bound coordinating organization called the Grid Forum (www.gridforum.org - [35]). From efforts such as this, Grids will become a reality, and an important component of the practice of science and engineering.

6 Acknowledgments

Many people have contributed to the work described here. Some of those people are identified in the text, and some in the citations. However, some other comments are necessary. Stewart C. Loken, head of the Information and Computing Sciences Division at LBNL has been a longtime supporter of this work, and as a experimental high energy physicist he has a special appreciation for its potential. Originally John Cavallini, and now Mary Anne Scott, of the DOE, Office of Science, computer science program office

(MICS) have been sufficiently convinced of the worth of Collaboratories to fund the related technologies for the past decade.

Many of the people who have worked on the MAGIC network testbed project (including the author) consider it to be one of the most successful of its type in contributing to the field of data intensive computing, and this is due not just to DARPA and DOE funding, but the many technically excellent people who worked on the project and toured the brew pubs of the midwest. Special thanks goes to Ira Richer, the long time project leader for MAGIC, with his ability to “herd cats” so effectively. Brian Tierney, Jason Lee, Gary Hoo, Jin Guojun, and Mary Thompson of LBNL have provided much of the implementation expertise to “make it all happen.”

Almost everyone in the NAS division of the NASA Ames Research Center, numerous other people at the NASA Ames, Glenn, and Langley Research Centers, as well as many people involved with the NSF PACIs [36] (especially Ian Foster, Argonne National Laboratory and Carl Kesselman, USC/ISI) are contributing to making the IPG vision a reality. In particular, we would like to thank Bill Feiereisen, NAS Division Chief, and, while the NASA HPCC Program Manager, the initiator of IPG; Alex Woo, NAS Research Branch Chief; Bill Thigpen, NAS Engineering Branch Chief, and; Bill Nitzberg and Dennis Gannon (Indiana University) “co-architects” with the author of IPG.

The work described in this paper was supported by the U. S. Dept. of Energy, Office of Science, Mathematical, Information, and Computational Sciences and ERLTT Divisions under contract DE-AC03-76SF00098 with the University of California, by DARPA, Information Technology Office, and by NASA’s Office of Aeronautics and Space Transportation Technology, Information Technology (IT) Program.

The author (see www-itg.lbl.gov/~johnston) may be reached as wejohnston@lbl.gov.

7 References and Notes

[1] Foster, I., and C. Kesselman, eds., *The Grid: Blueprint for a New Computing Infrastructure,* edited by Ian Foster and Carl Kesselman. Morgan Kaufmann, Pub. August 1998. ISBN 1-55860-475-8. http://www.mkp.com/books_catalog/1-55860-475-8.asp

[2] “Johnston, W. E., D. Gannon, and B. Nitzberg, “Grids as Production Computing Environments: The Engineering Aspects of NASA's Information Power Grid,” Eighth IEEE International Symposium on High Performance Distributed Computing, Aug. 3-6, 1999, Redondo Beach, California. (Available at http://www.nas.nasa.gov/~wej/IPG)

[3] “Information Power Grid.” See www.nas.nasa.gov/~wej/IPG for project information and pointers.

[4] Johnston, W., Lee, J., Tierney, B., and Tull, C., “Directions and Issues for High Data Rate Wide Area Network Environments,” Proceedings of the Computers in High Energy Physics Conference (CHEP 98), August 1998. (Available at http://www-didc.lbl.gov/publications.html .)

[5] Greiman, W., W. E. Johnston, C. McParland, D. Olson, B. Tierney, C. Tull, “High-Speed Distributed Data Handling for HENP,” Computing in High Energy Physics, April, 1997. Berlin, Germany. (Available at http://www-itg.lbl.gov/STAR .)

[6] NTON, “National Transparent Optical Network Consortium.” See http://www.ntonc.org.

[7] “Johnston, W., G. Jin, C. Larsen, J. Lee, G. Hoo, M. Thompson, and B. Tierney (LBNL) and J. Terdiman (Kaiser Permanente Division of Research). “Real-Time Generation and Cataloguing of Large Data-Objects in Widely Distributed Environments.” Invited paper, International Journal of Digital Libraries - Special Issue on “Digital Libraries in Medicine”. May, 1998. http://www-itg.lbl.gov/WALDO/

[8] Foster, I., C. Kesselman, Globus: A metacomputing infrastructure toolkit”, *Int’l J. Supercomputing Applications*, 11(2);115-128, 1997. (Also see http://www.globus.org)

[9] Public-Key certificate infrastructure (“PKI”) provides the tools to create and manage digitally signed certificates. For identity authentication, a certification authority generates a certificate (most commonly an X.509 certificate) containing the name (usually X.500 distinguished name) of an entity (e.g. user) and that entity’s public key. The CA then signs this “certificate” and publishes it (usually in an LDAP directory service). These are the basic components of PKI, and allow the entity to prove its identity, independent of location or system. For more information, see, e.g.,RSA Lab’s “Frequently Asked Questions About Today's Cryptography” http://www.rsa.com/rsalabs/faq/, *Computer Communications Security: Principles, Standards, Protocols, and Techniques*. W. Ford, Prentice-Hall, Englewood Cliffs, New Jersey, 07632, 1995, or *Applied Cryptography*, B. Schneier, John Wiley & Sons, 1996.

[10] Moore, R., et al, “Massive Data Analysis Systems,” San Diego Supercomputer Center. See http://www.sdsc.edu/MDAS

[11] Foster, I., N. Karonis, “A Grid-Enabled MPI: Message Passing in Heterogeneous Distributed Computing Systems.” Proc. 1998 SC Conference. Available at http://www-fp.globus.org/documentation/papers.html

[12] Grimshaw, A. S., W. A. Wulf, and the Legion team, “The Legion vision of a worldwide virtual computer”, *Communications of the ACM*, 40(1):39-45, 1997.

[13] Otte, R., P. Patrick, M. Roy, *Understanding CORBA*, Englewood Cliffs, NJ, Prentice Hall, 1996.

[14] Johnston, W. E., “High-Speed, Wide Area, Data Intensive Computing: A Ten Year Retrospective.” 7th IEEE Symposium on High Performance Distributed Computing, Chicago, Ill. July 29-31, 1998. Available at http://www-itg.lbl.gov/~johnston/papers.html

[15] Johnston, W., V. Jacobson, S. C. Loken, D. W. Robertson, and B. L. Tierney, “High-Performance Computing, High-Speed Networks, and Configurable Computing Environments: Progress Toward Fully Distributed Computing,” in *High-performance Computing in Biomedical Research*, T. Pilkington, et al, eds. CRC Press, 1993.

[16] Wiltzius, D., L. Berc, and S. Devadhar, “BAGNet: Experiences with an ATM metropolitan-area network,” ConneXions, Volume 10, No. 3, March 1996. Also see http://www.llnl.gov/bagnet/connexions.html .

[17] LBNL Network Research Group, “vic, vat, wb, and sd,” the MBone (IP multicast) teleconferencing tools, described at http://ee.lbl.gov.

[18] Johnston, W.,G. Jin, C. Larsen, J. Lee, G. Hoo, M. Thompson, and B. Tierney (LBNL) and J. Terdiman (Kaiser Permanente Division of Research), “Real-Time Generation and Cataloguing of Large Data-Objects in Widely Distributed Environments.” Invited paper, International Journal of Digital Libraries - Special Issue on “Digital Libraries in Medicine”. May, 1998. http://www-itg.lbl.gov/WALDO/

[19] Tierney, B. Lee, J., Crowley, B., Holding, M., Hylton, J., Drake, F., “A Network-Aware Distributed Storage Cache for Data Intensive Environments”, Proceeding of IEEE High Performance Distributed Computing conference (HPDC-8), August 1999.

[20] Fuller, B., I. Richer “The MAGIC Project: From Vision to Reality,” IEEE Network, May, 1996, Vol. 10, no. 3.

[21] MAGIC, “The MAGIC Gigabit Network”, See: http://www.magic.net

[22] Lau, S, and Y. Leclerc, “TerraVision: a Terrain Visualization System,”, Technical Note 540, SRI International, Menlo Park, CA, Mar. 1994. (http://www.ai.sri.com/~magic/terravision.html)

[23] Brooks, C., B. Tierney, W. Johnston, “Java Agents for Monitoring and Management (JAMM)”. JAMM is a heterogeneous system of brokers and agents that manage components, access to, and information about a Distributed Parallel Storage System (DPSS) and its associated clients. JAMM brokers and agents are designed to be general problem-solving entities. That is, rather than having a predefined set of tasks they can perform, they are able to perform goal-based reasoning and inference to determine a solution to a specific problem from general facts and rules. They are also able to add new facts and rules to their knowledge base, allowing them to be dynamically extended. See http://www-didc.lbl.gov/JAMM/ .

[24] Livny, M, et al, “Condor.” “Condor is a High Throughput Computing environment that can manage very large collections of distributively owned workstations. Its development has been motivated by the ever increasing need of scientists and engineers to harness the capacity of such collections. The environment is based on a novel layered architecture that enables it to provide a powerful and flexible suite of Resource Management services to sequential and parallel applications. Condor is currently available on many UNIX platforms. An effort to port Condor to Windows-NT is currently underway. Condor has been used in production mode for more than 10 years in our department and around the world.” See http://www.cs.wisc.edu/condor/

[25] Sun Microsystems, “Remote Method Invocation (RMI).” See http://developer.java.sun.com/developer/technicalArticles//RMI/index.html .

[26] Microsoft Corp., “DCOM Technical Overview.” November 1996. See http://msdn.microsoft.com/library/backgrnd/html/msdn_dcomtec.htm .

[27] “XML Metadata Interchange” (XMI). “The main purpose of XMI is to enable easy interchange of metadata between tools and metadata repositories (based on OMG MOF) in distributed heterogeneous environments. The major initial use of XMI will be to interchange UML models between modeling tools (based on the OMG UML) and repositories (based on OMG MOF and UML).” See “XML News and Resources” http://metalab.unc.edu/xml/

[28] Fitzgerald, S., I. Foster, C. Kesselman, G. von Laszewski, W. Smith, S. Tuecke, “A Directory Service for Configuring High-Performance Distributed Computations.” S. Fitzgerald, Proc. 6th IEEE Symp. on High-Performance Distributed Computing, pg. 365-375, 1997. Describes the Metacomputing Directory Service - now called the Grid Information Service - used to maintain information about Globus components. Available from http://www-fp.globus.org/documentation/papers.html .

[29] "Bridging the Gap from Networking Technologies to Applications." Workshop Co-sponsored by HPNAT & NRT (High Performance Network Applications Team & Networking Research Team of the Large Scale Networking (Next Generation Internet) Working Group). NASA Ames Research Center, Moffett Field, Mountain View CA. Moffett Training and Conference Center, August 10 - 11, 1999. To be published at http://www.nren.nasa.gov/workshop_home.html ("HPNAT/NRT Workshop")

[30] Parvin, B., D. E. Callahan, W. Johnston, and M. Maestre, "Visual Servoing for Micro Manipulation," International Conference on Pattern Recognition, August. 1996. (Available at http://www-itg.lbl.gov/ITG.hm.pg.docs/VISION/vision.html)

[31] Parvin, B., J. Taylor, D. E. Callahan, W. Johnston, and U. Dahmen, "Visual Servoing for Online Facilities," IEEE Computer July 1997. (Available at http://www-itg.lbl.gov/ITG.hm.pg.docs/VISION/vision.html)

[32] Agarwal, D., at al, "The Spectro-Microscopy Collaboratory at the Advanced Light Source." See http://www-itg.lbl.gov/BL7Collab .

[33] Johnston, W., et al, "The Distributed, Collaboratory Experiment Environments (DCEE) Program." http://www-itg.lbl.gov/DCEE

[34] "Globus Ubiquitous Supercomputing Testbed Organization" (GUSTO). At Supercomputing 1998, GUSTO linked around 40 sites, and provides over 2.5 TFLOPS of compute power, thereby representing one of the largest computational environments ever constructed at that time. See http://www.globus.org/testbeds .

[35] Grid Forum. The Grid Forum (www.gridforum.org) is an informal consortium of institutions and individuals working on wide area computing and computational grids: the technologies that underlie such activities as the NCSA Alliance's National Technology Grid, NPACI's Metasystems efforts, NASA's Information Power Grid, DOE ASCI's DISCOM program, and other activities worldwide.

The Grid Forum is modeled, in many respects, on the Internet Engineering Task Force (IETF) and focuses on the promotion of Grid computing via the documentation of "best practices" and "standards", with an emphasis on rough consensus and running code. The processes under which the Grid Forum are still being established and will be discussed at the next meeting.

The work of the Grid Forum is performed within its various working groups. The following working groups have been established to date:

- Scheduling Working Group (Sched-WG)
- Grid Information Service Working Group (GIS-WG)
- Security Working Group (Security-WG)
- Remote Data Access Working Group (Data-WG)
- Application and Tools Requirements Working Group (Apps-WG)
- End-to-end Performance Working Group (Perf-WG)
- Advanced Programming Models Working Group (Models-WG)
- Account Management Working Group (Accounts-WG)
- User Services Working Group (Users-WG)

[36] The NSF PACIs are the Alliance/NCSA (http://www.ncsa.uiuc.edu/) and NPACI/SDSC (http://www.npaci.edu/).